diffusion simulations: work-in-progress

V. Previtali

special thanks to D. Shatilov

diffusion studies

- diffusion studies: imposing an initial distribution and checking how the distribution of single-particle emittances "naturally" change after a large turn number
- "naturally" means considering the contribution of different effects:
  - beam-beam when in collision (for the collider mode)
  - beam-gas interaction and RF noise
  - lattice non linearities
  - other effects will be implemented (e.g. IBS, space charge...)
  - no interaction with collimators is considered at the moment (required?)
- the goal is: calculate the diffusion coefficient for different particle amplitudes. This coefficient can be later compared with measurements (derived by collimator baby-step measurements, Giulio)

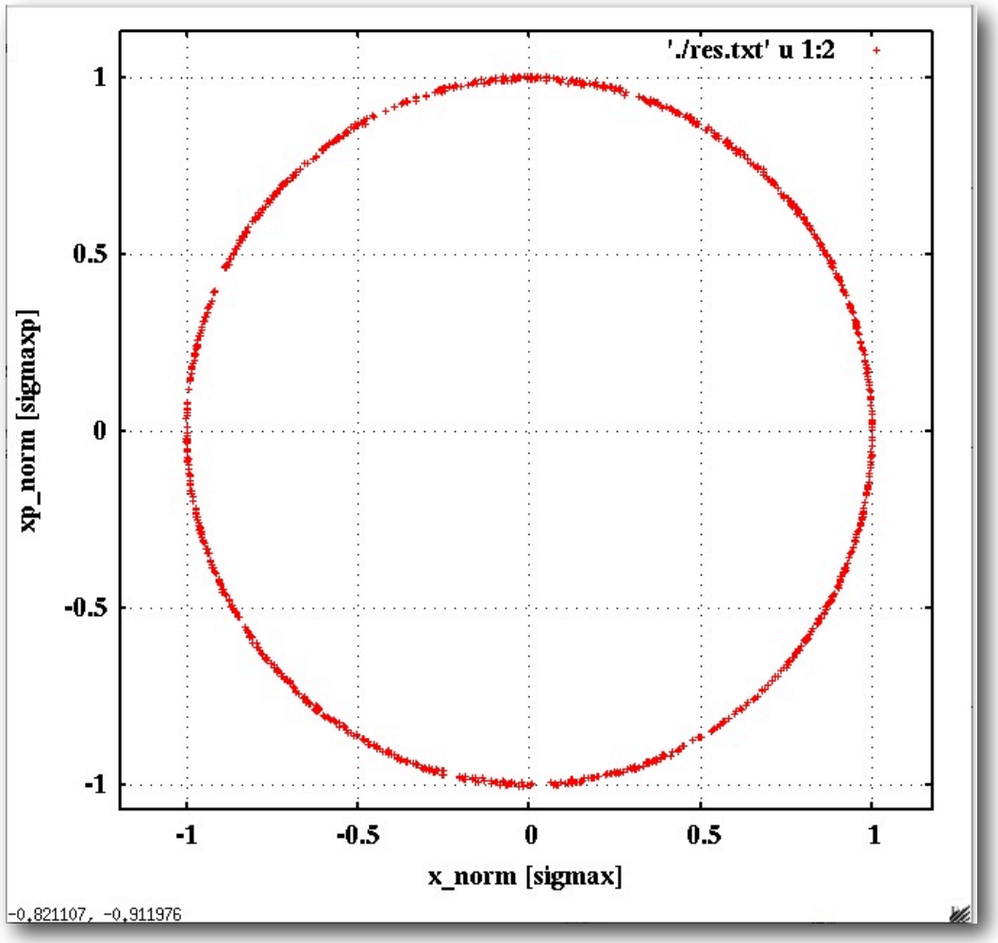
working with normalized coordinates

- we are interested in measuring the change in emittance
- to calculate particle emittance, we need to work in normalized coordinates
- lifetrack: the distribution ca be read from an input file.
  - It accepts both normalized and physical coordinates as an input (ascii file).
  - the output is only in physical coordinates (binary file).
- the output files must be normalized => a normalization matrix is need
  - since the Tevatron is coupled (both between the transversal and longitudinal planes) the usual courant-snyder normalization is not valid
  - by calculating the 6D one-turn matrix eigenvectors, it is possible to get the normalization matrix
  - *Thanks to Dmitry*, now the normalization matrix can be given in the standard output of Lifetrack, when it is launched with the -deb option

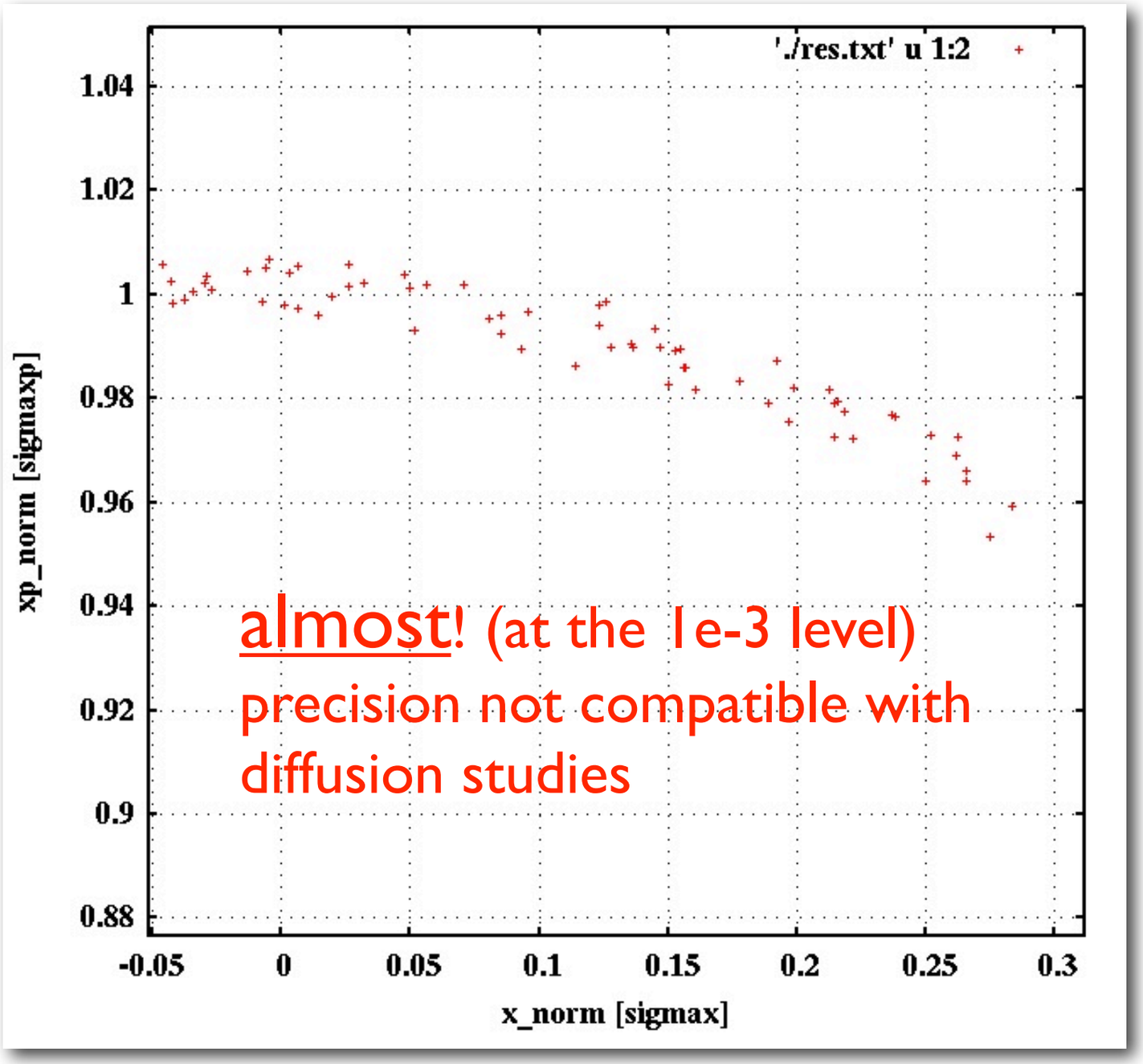
normalization TEST

- To check the normalization, I run a test job for initial distribution:
  - 1 sigma *x*
  - 1sigma *y*
  - exactly (0,0) in the longitudinal coordinates (*z*)
- With *x, y, z* we refer here to the skew oscillation modes of our one-turn matrix at the observation point. In these planes the emittances are conserved (for a linear machine).
- PROCEDURE:
  1. give the described initial distribution to lifetrack (1000 particles)
  2. read the coor.0 output file that should correspond to the physical coordinates of the initial distribution
  3. normalized the coor.0 coordinates with the normalization matrix
  4. check that the input and output distribution are the same

first results: Octave approach



transverse plane



almost! (at the 1e-3 level)
precision not compatible with diffusion studies

the observed spread was given by loss in numerical precision

1- when translating from binaries (fixed)

2- importing the elements of the normalization matrix in octave

transverse plane

improving the numerical precision

- The numerical precision of the final normalized coordinates are obviously related to the numerical precision of :

1. the initial distribution file that I provide to lifetrack
2. the numbers stored in the coor.0 file
3. the number in the “translated” coor.0 file
4. the normalization matrix elements

transverse plane

improving the numerical precision

- The numerical precision of the final normalized coordinates are obviously related to the numerical precision of :

1. the initial distribution file that I provide to lifetrack ➡ initially %.6f
2. the numbers stored in the coor.0 file ➡ double (16 sign. digits)
3. the number in the "translated" coor.0 file ➡ initially %.6f
4. the normalization matrix elements ➡ initially %.4f (OCTAVE approximation)

transverse plane

improving the numerical precision

- The numerical precision of the final normalized coordinates are obviously related to the numerical precision of :

1. the initial distribution file that I provide to lifetrack ➡ now %.8f
2. the numbers stored in the coor.0 file ➡ double
3. the number in the "translated" coor.0 file ➡ double
4. the normalization matrix elements ➡ > %.10f (ltr output)

✓ written a dedicated c code to automatically read and normalize the coor files.

transverse plane

improving the numerical precision

- The numerical precision of the final normalized coordinates are obviously related to the numerical precision of :

1. the initial distribution file that I provide to lifetrack ➡ now %.8f
2. the numbers stored in the coor.0 file ➡ double
3. the number in the "translated" coor.0 file ➡ double
4. the normalization matrix elements ➡ > %.10f (ltr output)

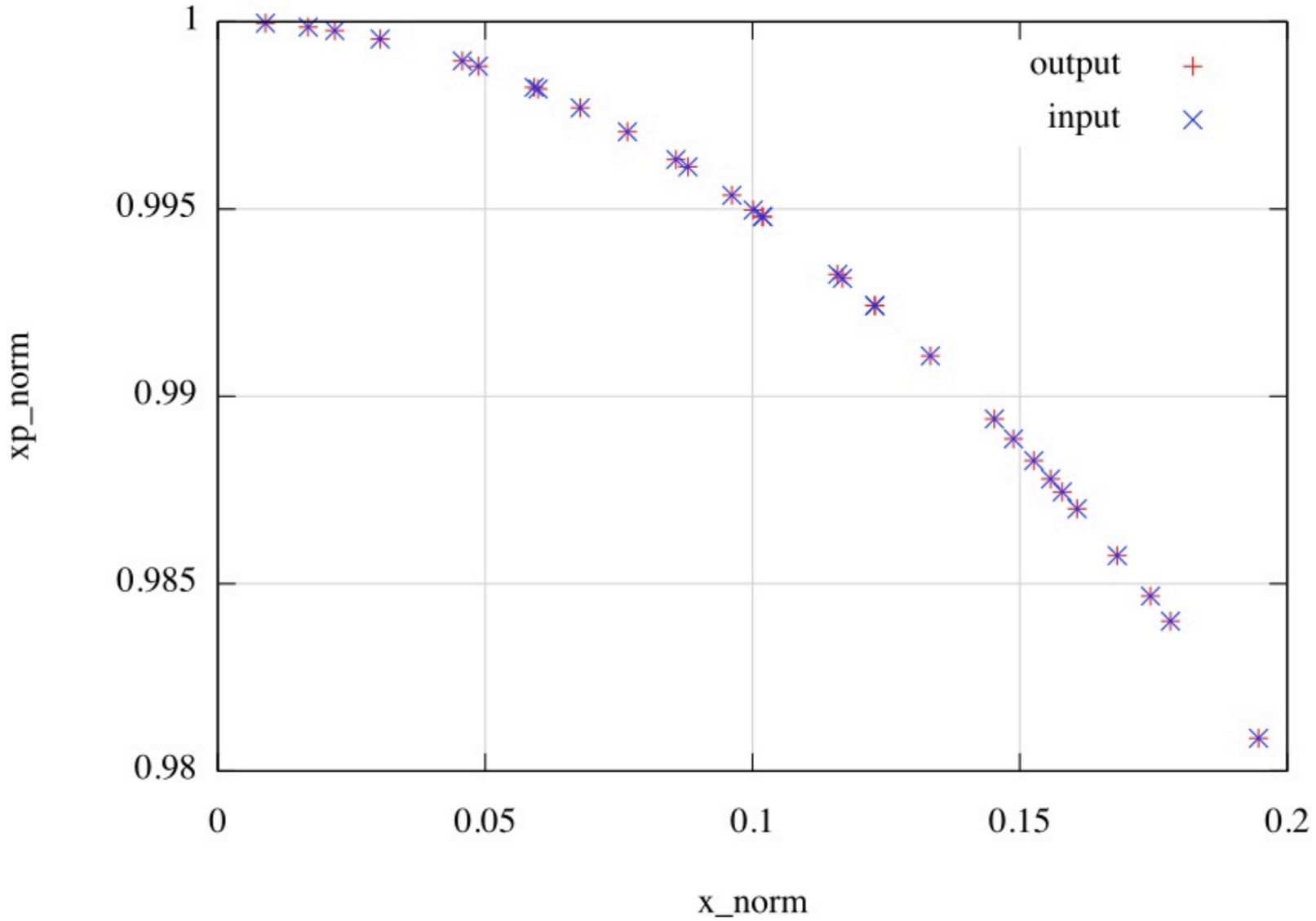
lifetrack does not accepts inputs with numerical precision larger than %.8f OR %.4e (?)

transverse plane

achieved numerical precision

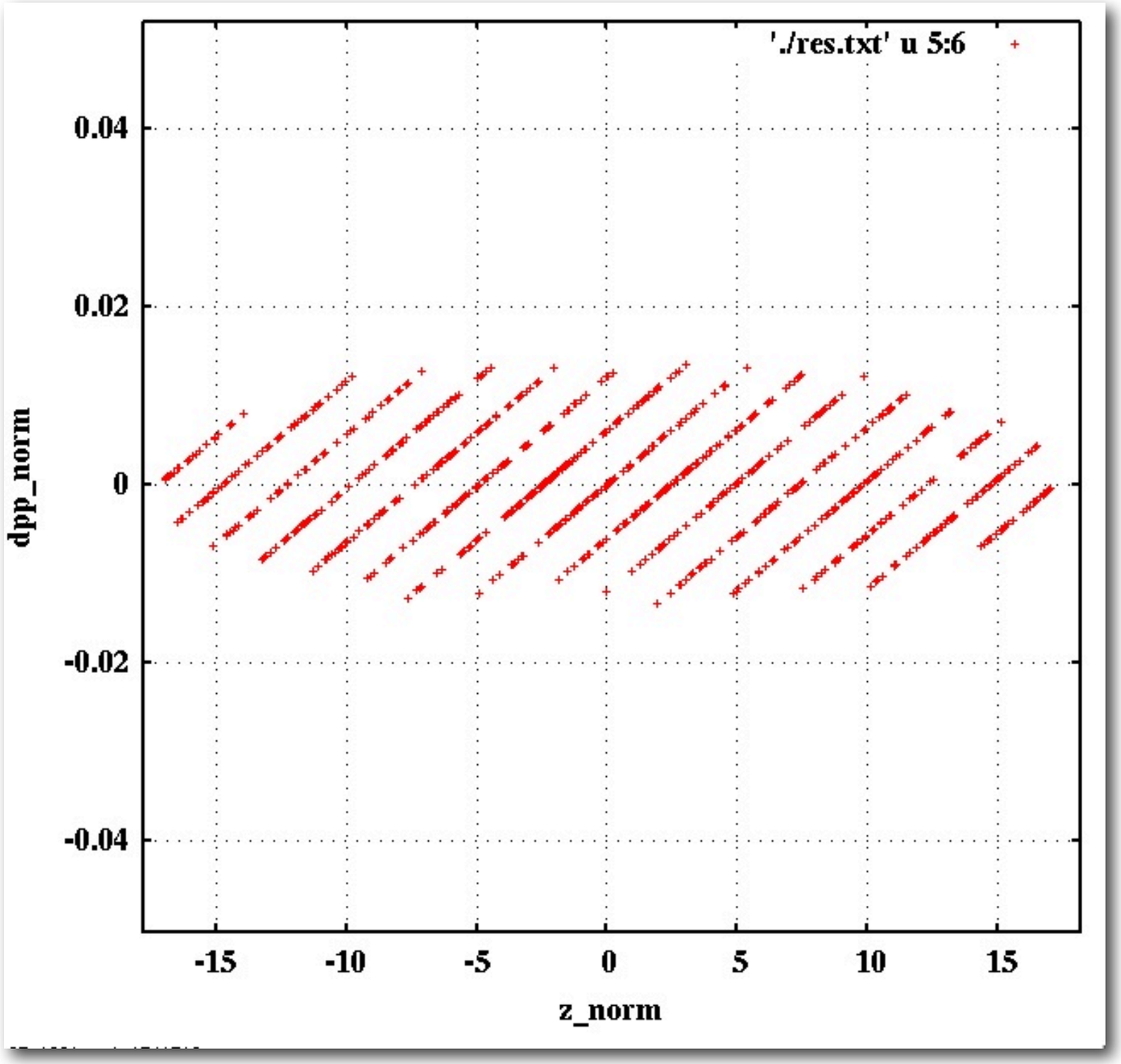
- the distance between input and normalized output coordinates is <=1e-10

- the radius in the normalized phase space is now 1-1e-8<r<1+1e-8



longitudinal plane

first results for longitudinal plane



differences at the 1e-2 level

non acceptable difference between input and output file

non deriving from numerical precision issues

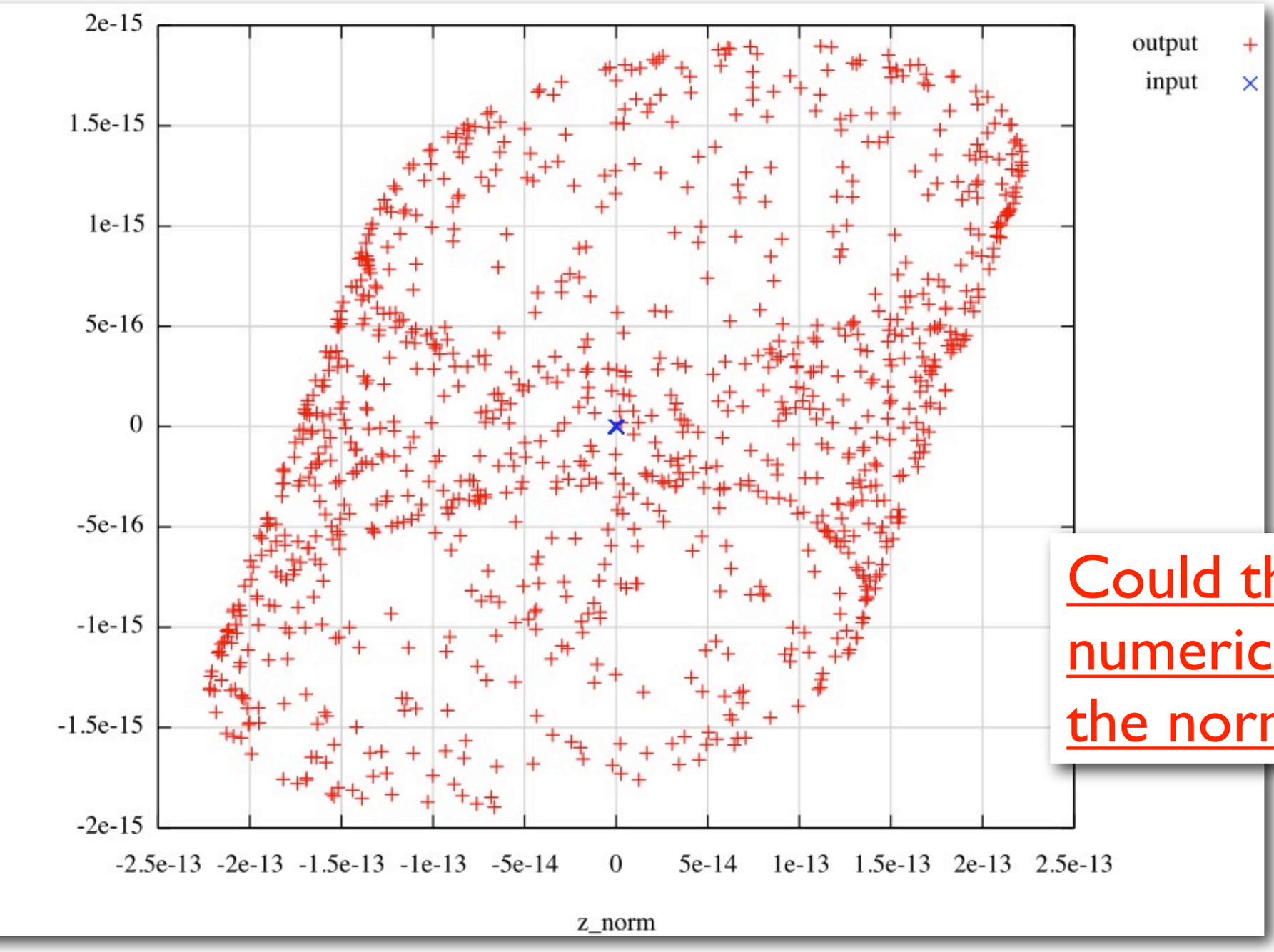
lifetrack peculiarity: it used to re-adjust the physical coordinates of the longitudinal distribution to match the average (dE,dz) given in the .tsk file

change in the code

"Now it is fixed in such a way that, if the initial distribution is given explicitly (a Distr_init file is provided) re-normalization of longitudinal coordinates is not performed. Otherwise, the distribution is assumed to be Gaussian and re-normalization is performed. " [Dmitry]

longitudinal plane

achieved numerical precision



with the new version of Lifetrack, the distance between input and output normalized coordinates is less than 1e-13

Could the "problem" be the numerical approximation of the normalization matrix?

new updated code

- Thanks to the fast intervention of Dmitry, the new updated lifetrack code is available on the Wilson cluster, in */home/shatilov/lifetrac/src_3.37/*
- *question to Dmitry/Sasha:* is also lifetrap (for parallel simulations)updated accordingly?

next steps

- Define the distributions: ring of n sigmax. What about sigmay? And dpp?
  - ✓ probably is better to keep same amplitude in both transverse planes. (done, for nsigma = 2.00 .. 0.2 .. 10.00, gaussian in normalized radius)
- Define the simulation cases
  - test: no random noise and beam beam.
    - ✓ input prepared. 10k particles, 300 steps/10k turns each
  - storage ring mode: random noise but no beam beam. What are the appropriate values for the random kicks?
  - which effects must be added? IBS and? (space charge negligible?)
- Collimators?